ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1er Recuperatorio
Fecha examen: viernes 13 / Fecha notas: 17-JUL-2018

	Nº Orden	Apellido y nombre	L.U.	Cant. hojas[1]
Completar:	315			8
	Nota (Nº)	Nota (Letras)	Docente	
No completar:	7	siete (Aprobado)	Gonzalo	

1. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2. Se dice que un grafo J es un grafo junta si y sólo si existen grafos G_1 y G_2 tales que $J = G_1 + G_2$.

   2

   Demostrar que J es un grafo junta si y sólo si J^c es no conexo.

2. Alicia, un burro y un conejo quieren ir desde El País de las Maravillas hasta El Palacio de las Zanahorias. Para hacerlo deben cruzar un puente colgante que resiste el peso de a lo sumo dos individuos. Además, como es de noche, cualquiera que cruce el puente debe llevar una linterna. Alicia y sus amigos tienen una única linterna. Si Alicia cruza sola el puente, tarda 5 minutos, mientras que el burro tarda 2 minutos, y el conejo 1 minuto. Si dos individuos cruzan el puente juntos, tardan lo que tardaría el más lento de los dos si fuera solo, debido a que deben usar la linterna para iluminar el recorrido de ambos individuos. Por ejemplo, si Alicia y el conejo cruzan juntos el puente, tardan 5 minutos en hacerlo. Alicia, el burro y el conejo quieren llegar lo antes posible a destino. Modelar la situación como un problema de camino mínimo en un grafo, y resolverlo por el método de Dijkstra. Justificar.

   0

3. Sea G un grafo conexo con pesos asociados a sus ejes. Sea $e_{\min}(G)$ cualquier eje de G que tenga peso mínimo y $e_{\max}(G)$ cualquier eje de G que tenga peso máximo. ¿Es cierto que...

   2

   (1)
   (a) $e_{\min}(G)$ pertenece a algún árbol generador mínimo de G? 0.2
   (b) $e_{\min}(G)$ pertenece a todo árbol generador mínimo de G? 0.2
   (c) $e_{\max}(G)$ no pertenece a algún árbol generador mínimo de G? 0.2
   (d) $e_{\max}(G)$ no pertenece a ningún árbol generador mínimo de G? 0.2

   Demostrar o dar un (contra) ejemplo y justificar según corresponda.

   (2) Repetir suponiendo que existe un único eje de peso mínimo y un único eje de peso máximo.

4. 1
   (a) Sea G un grafo de n vértices, d-regular y autocomplementario. Demostrar que $n = 2d + 1$ y d es par. 0.
   SUGERENCIA: Usar que un grafo es regular si y sólo si su complemento lo es.
   (b) Sea un entero positivo $n \not\equiv 1 \pmod 4$. Demostrar que hay una cantidad par de grafos regulares de n vértices. 0.
   (c) Para cada entero positivo $n \leq 7$ indicar cuántos grafos regulares de n vértices hay, y cuántos de ellos son autocomplementarios. Justificar. 0.
   SUGERENCIA: Para cada n considerar cada posible d y armar una tabla.

5. Viajante de comercio bitónico: Se tienen $n \geq 2$ puntos en el plano $p_1, p_2, \ldots p_n$, ordenados de manera creciente de acuerdo a su coordenada x (no hay dos puntos con la misma coordenada x). Un recorrido bitónico de los puntos es un recorrido que comienza en p_1, recorre de manera creciente en x algunos de los puntos hasta llegar a p_n (ida), y finalmente recorre de manera decreciente en x algunos de los puntos hasta volver a p_1 (vuelta); el recorrido pasa exactamente una vez por cada punto.

   2

   Para $n = 2$ hay un único recorrido bitónico, que es p_1, p_2, p_1. Para $n = 3$ hay dos recorridos bitónicos. Uno es p_1, p_2, p_3, p_1, mientras que el otro es p_1, p_3, p_2, p_1. Sin embargo, ambos recorridos tienen la misma longitud, ya que lo que recorre uno a la ida, lo recorre el otro a la vuelta, y viceversa. En general, dado cualquier recorrido bitónico, existe uno "simétrico" con la ida y la vuelta intercambiadas.

   Diseñar un algoritmo que encuentre la mínima longitud que puede tener un recorrido bitónico. El algoritmo debe tener complejidad $O(n^2)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n^2)$ y espacial $O(n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

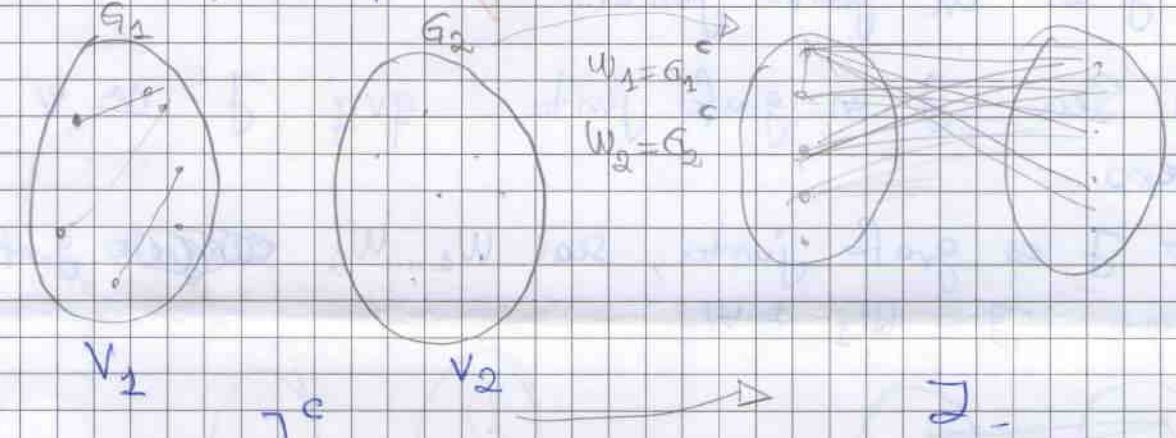
   SUGERENCIA: Para $i \neq j$, sea $f(i, j) = f(j, i)$ la longitud mínima para completar hasta p_1 un recorrido bitónico a partir de una porción que va de p_i a p_n, vuelve de p_n a p_j, y pasa exactamente una vez por cada punto $p_{\min\{i,j\}}, p_{\min\{i,j\}+1}, \ldots p_n$. La solución es $f(n-1, n) + d(n-1, n)$, donde $d(i, j) = d(j, i)$ es la distancia (en el plano) entre p_i y p_j.

[1] Incluyendo a esta hoja. Entregar esta hoja junto al examen.

Ejercicio 1 = Probar que J es un grafo junta $\iff$ J^c es no conexo.

Dem =

$\Leftarrow$) Sea J grafo tal que J^c no es conexo. qvq J es grafo junta.
Como J^c no es conexo existen $V_1, V_2 \in V$, $V_1, V_2 \neq \emptyset$ tq $V_1 \cap V_2 = \emptyset$ y $V_1 \cup V_2 = V$.



Afirmo que $J = W_1 + W_2$, o sea J es la junta de W_1 y W_2 ~~donde W_1 es el subgrafo que se forma con los vértices de V_1 y sus respectivas aristas y W_2 lo mismo pero con V_2.~~
donde W_1 y W_2 son los grafos complementos de los subgrafos que se forman en J^c con los vértices de V_1 y V_2 respectivamente y sus respectivas aristas. ($W_1 = G_1^c$ y $W_2 = G_2^c$)
Veamos que esta afirmación es cierta.
Como G_1 y G_2 son subgrafos de J^c y no hay aristas que conecten nodos de G_1 con nodos de G_2 (pues J^c no es conexo) entonces $G_1^c = W_1$ y $G_2^c = W_2$

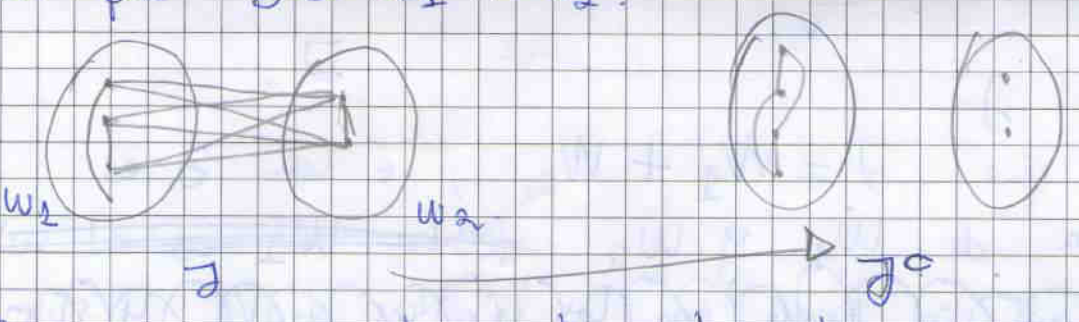
Son subgrafos de J (pues el complemento de un grafo tiene los mismos nodos que el grafo pero tiene "las aristas invertidas"). Además en J están todas las aristas entre todos los nodos de G_1 y todos los nodos de G_2, pues en J^c no estaban. Esto significa que J es la junta de los grafos G_1^c y G_2^c, es decir,

$J = W_1 + W_2$.

$\therefore$ J es un grafo junta.

$\Longrightarrow$) Sea J un grafo junta. qvq J^c no es conexo.

Como J es grafo junta, sean W_1, W_2 grafos tal que $J = W_1 + W_2$.



Con un razonamiento similar al anterior, y como se trata de grafos y sus complementos, todas las aristas "de la junta" de W_1 con W_2 que están en J, en J^c no van a estar, o sea tengo una desconexión de J^c, pues como la junta conecta todos los nodos de W_1 con todos los nodos de W_2 y viceversa y esas conexiones en J^c no van a estar, no hay manera en J^c de armar un camino que conecte a un nodo de W_1 con un nodo de W_2, por lo tanto J^c no es conexo.

Mejor.

(También podría haber armado una partición de $J^c = W_1^c \cup W_2^c$ y con esto me quedan al menos 2 comp. conexas lo cual me dice q' J^c no es conexo)

Asamblea

Ejercicio 3 = G conexo y con pesos en los ejes.
$e_{min}(G)$ = cualquier eje de peso mín en G, $e_{max}(G)$ = eje de peso máx.

1 - a - $e_{min}(G)$ pertenece a algún AGM de G?

VERDADERO.

Supongamos que $e_{min}(G)$ no está en todos los AGM de G.

Sea T un AGM de G cualquiera.

Considero $T + e_{min}(G)$, grafo en el cual hay un único ciclo simple y $e_{min}(G)$ está en él, pues T es árbol.

Sea $f \in T$, f arista tal que f pertenece al ciclo simple donde está $e_{min}(G)$. en $T + e_{min}(G)$.

Considero $T_1 = T + e - f$. Como T era AGM, en particular AG entonces T_1 es un AG de G, Prop. vista en la práctica; cuál?

. Si $peso(e_{min}(G)) = peso(f) \Rightarrow peso(T_1) = peso(T)$
$\Rightarrow T_1$ es un AGM de G Abs! pues $e_{min}(G)$ no estaba en ningún AGM

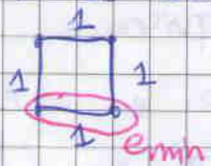
. Si $peso(e_{min}(G)) > peso(f)$ NO puede pasar pues $e_{min}(G)$ es la arista de peso mínimo en G.

. Si $peso(e_{min}(G)) < peso(f) \Rightarrow peso(T_1) = peso(T + e_{min} - f)$
$\Rightarrow peso(T_1) = peso(T) + \underbrace{peso(e_{min}) - peso(f)}_{<0}$
$\Rightarrow peso(T_1) < peso(T)$ ABS! pues T es AGM!

$\Rightarrow e_{min}(G)$ pertenece a algún AGM de G

2-a- Si existe un único eje de peso mínimo:
a - Es Verdadero. Idem demostración que antes, sólo que no puede pasar que haya otras aristas con igual peso que e_{min}. ✓

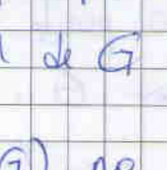
1-b- e_{min} pertenece a todo AGM? FALSO.

Sea $G =$  Sea $T =$ 

T es AGM de G y $e_{min} \notin T$.

2-b- Si existe un único eje de peso mínimo: entonces tiene que estar en <u>todos</u> los AGM de G.
Supongamos que e_{min} <u>no</u> está en todos los AGM
$\Rightarrow$ existe algún AGM, T, tal que e_{min} <u>no</u> está en él. Con el mismo razonamiento que en el ítem 1.a- y considerando q' e_{min} es el único eje de menor peso, llegamos a que existe un AG de G de menor peso que T lo cual es ABS pues T era un AGM de G (ie, AG de peso mínimo). ✓

1-d- $e_{max}(G)$ <u>no</u> pertenece a ningún AGM de G?
FALSO.

Sea $G =$  como G es un árbol, i.e. conexo y sin ciclos, el único AGM de G es G ∇ y por lo tanto $e_{max}(G)$ está en todos los AGM de G.

2-d- El ejemplo anterior sirve para ver que 2-d es Falso también. ✓

hay algún AGM en el cual no pertenece

1 - c - $e_{max}(G)$ no pertenece a algún AGM de G ?
FALSO.
El grafo del ítem 1-d sirve como contraej. pues $e_{máx}$ está en todos los AGM de G.

2 - c - FALSO. El mismo ejemplo que en 1-d sirve.

OBSERVACIÓN = Si G es árbol ⇒ todos los AGM de G son el conjunto {G},
Pues como un AGM tiene q' ser un árbol generador ⇒ G es el único árbol generador de G (valga la redundancia ☺), y como es el único, es el mínimo.

Ejercicio 5 = Viajante de comercio bitónico:

$P_1, \dots, P_n$ puntos en el plano, $n \geq 2$ y ordenados de manera crec. con respecto a la coord x. (y además no hay 2 puntos con la misma coord x).

Algoritmo que encuentre la mínima long. que puede tener un recorrido bitónico.

Consideremos la función:

$f(j,i) = f(i,j)$ = "la mínima longitud de un camino que vuelve desde p_i a p_1 ($P_1 \leftarrow P_i$) y va desde P_1 a p_j ($P_1 \longmapsto P_j$) * ($i \neq j$) pasando exactamente una vez por cada punto $P_1, \dots, P_{máx\{i,j\}}$."

Si tenemos esta función definida lo que queremos hacer para tener la solución de nuestro problema es minimizar $f(i,n) + d(p_i, P_n)$, con $1 \leq i < n$

Pues $f(i,n)$ me va a dar la long. mínima de un camino $P_1 \leftarrow P_i$, $P_1 \longmapsto P_n$

$\underbrace{P_1 \dots P_n}\underbrace{P_i \dots P_1}$

la long mín de esta parte del camino me la dá $f(i,n)$

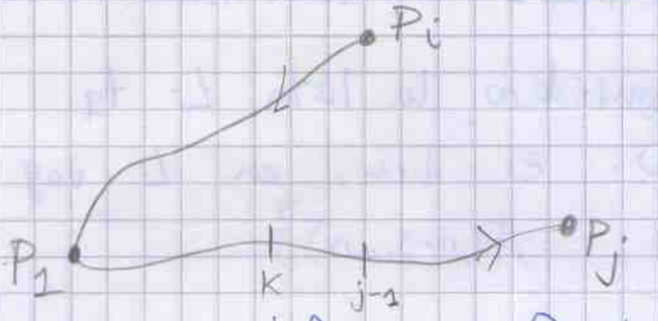
pero para que sea un recorrido bitónico, falta agregarle la distancia de P_n a P_i.

Luego, la solución al problema es:

$$\min_{1 \leq i < n} \{ f(i,n) + d(P_i, P_n) \}.$$

FECHA

Vamos a definir $f(i,j)$. Consideremos que $j > i$ (Si $i > j$ la definición de f sería análoga dado que en nuestro problema lo que representa $f(i,j)$ $\min Long(P_i \mapsto P_1 \mapsto P_j)$ es lo mismo que lo que representa $f(j,i)$ $\min Long(P_j \mapsto P_1 \mapsto P_i)$).



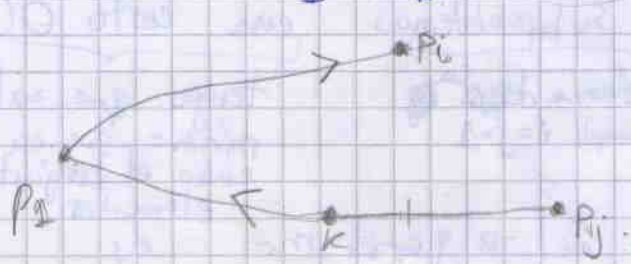
Queremos definir a f basándonos en "sub instancias" de la misma, i.e. instancias más chicas.

Consideremos 2 casos:

▲ Si $i \neq j-1$: $f(i,j) = f(i,j-1) + d(P_{j-1}, P_j)$

▲ Si $i = j-1$: Sea $1 \leq k < i$, donde P_k representa el "punto anterior" a P_j en el recorrido desde P_1 a P_j ($P_1 \mapsto P_k \mapsto P_j$).

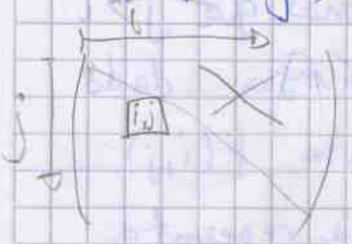
$$f(i,j) = \min_{1 \leq k < i} \{ f(k,i) + d(P_k, P_j) \}$$



Muy bien.

NOTA

Suponiendo $j > i$



$$f(i,j) = \begin{cases} d(p_i, p_j) & si \quad i=1 \wedge j=2 \\ f(i,j-1)+d(p_{j-1},p_j) & si \quad i\neq j-1 \\ \min\limits_{1\le k<i} \{\underbrace{f(k,i)}_{=f(k,j-1)} + d(p_k,p_j)\} & si \quad i=j-1 \end{cases}$$

Idea del algoritmo que resuelve el problema:

1- Calcular $f(i,j)$ y guardarlo en una matriz D de $(n-1)\times n$

2- Considero la lista L tq L = n-ésima FILA de D. Es decir, en L voy a tener $\{f(1,n), f(2,n), \dots, f(n-1,n)\}$. $O(n)$

3- Calculo el mínimo de la lista L. $O(n)$

4- Devuelvo ese mínimo calculado en 3. $O(1)$

- Veamos la complejidad temporal y espacial del algoritmo:

- <u>temporal</u> = $O(\#\text{subproblemas} \cdot \text{costo de calcularlos})$
- $i \neq j-1$: tengo $O(n^2)$ subproblemas con costo $O(1)$
  - busco en la matriz $O(1)$
  - calculo distancia (p_{j-1}, p_j) $O(1)$
- $i = j-1$: tengo $O(n)$ subproblemas con costo $O(n)$
  - sólo una columna de la matriz D cumple $i=j-1$
  - tengo que calcular un mínimo que en el peor caso el conjunto tiene n elementos.

$\therefore$ La complejidad temporal del algoritmo es $O(n^2)$.

NOTA

7/8

• espacial: $O(n^2)$ pues tengo que llenar toda la matriz D.

Haciendo una pequeña modificació al algoritmo vamos a conseguir complejidad espacial $O(n)$.

Dado que para calcular $f(i,j)$ en el peor caso tenemos que calcular un mínimo de n-elementos, pero que ésos elementos están todos en la misma FILA, la FILA $j-1$, y en el otro caso también sólo miro la FILA $j-1$ para calcular $f(i,j)$, en vez de guardar toda la matriz, sólo voy a ir guardando una lista con la FILA anterior. Esto me da complejidad espacial de $O(n)$.

¡Perfecto

Ejercicio 4 =

a – G grafo d-regular y $\overbrace{\text{auto complementario}}^{G \text{ y } G^c \text{ son iso}}$

Probar que $n = 2d + 1$ y d es par.

Como G es d-regular entonces G^c es $n-1-d$ regular.

Como G es auto complementario, es decir, G y G^c son isomorfos entonces, por ej. 5.19.c G y G^c tienen la misma cantidad de nodos de grado r, para $r \geq 0$.

Por lo tanto G es $n-1-d$ regular y G^c es d-reg. es decir $n - 1 - d = d$

$\therefore\ n = 2d + 1$

- Sup. que d es impar, No hace falta ir por el absurdo

$$\Rightarrow \quad 2m = \sum_{i=1}^{n} d(v_i) \underset{\substack{\downarrow \\ G \text{ es d-reg.}}}{=} \sum_{i=1}^{n} d = n \cdot d.$$

$2m = n \cdot d$ como

d impar $\Rightarrow$ $n = 2d + 1$ impar.

$\Rightarrow$ $n \cdot d$ producto de impares, es impar. ABS! pues $2m$ es PAR.

$\therefore$ d es par.

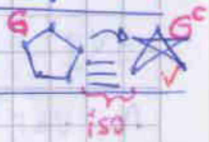
b – Sea un entero positivo $n \not\equiv 1 \pmod 4$

Probar que hay una cantidad par de grafos regulares de n vértices.

CONTINÚA EN LA SIGUIENTE PÁGINA. (*)

FECHA

C. Para cada $n \leq 7$ indicar cuántos grafos regulares de n vértices hay y cuántos de ellos son autocomplementarios.

$d \rightarrow$ / $n \downarrow$	0	1	2	3	4	5	6	$\boxed{\text{total}}$	Autocomplementario	
1	1	-	-	-	-	-	-	$\rightarrow$ 1	1	K_1
2	1	1	-	-	-	-	-	$\rightarrow$ 2	0	
3	1	-	1	-	-	-	-	$\rightarrow$ 2	0	
4	1	1	1	1	-	-	-	$\rightarrow$ 4	1	
5	1	-	1	-	1	-	-	$\rightarrow$ 3	1	
6	1	3	2	2	3	1	-	$\rightarrow$ 12	0	
7	1	-	2	-	2	-	1	$\rightarrow$ 6	0	

Esta tabla no fue escrita mágicamente sino que fue realizada teniendo en cuenta todos los siguientes puntos: Ok, pero ¿cómo?

1- Si $n \leq d$ no hay grafos d-reg de n vért.
2- Si $d = n-1$ el único grafo d-reg es K_n.
3- Si $d = 0 \Rightarrow$ todas las comp. conexas de un grafo G d-reg son $K_1 \Rightarrow \forall\, n \leq 7$, hay un sólo grafo 0-regular. (todos puntos aislados).
4- Si $d = 1 \Rightarrow$ todas las comp. conexas de G son K_2
5- Si $d = 2 \Rightarrow$ " " " " " son ciclos simples.
6- Si G es d-reg $\Leftrightarrow G^c$ es $n-d-1$ reg
7- Si n es impar y d es impar $\Rightarrow \nexists$ grafo d-regular de n-vértices $\left(\underbrace{2m}_{\text{PAR}} = \sum_{i=1}^{n} d(v_i) = \underbrace{n \cdot d}_{\text{impar}} \text{ ABS!} \right)$

NOTA

8 - G de n vert. d-reg y tq d es impar
$\Rightarrow$ G no es autocomplementario.

9 - G de n-vértices d-reg, d par y
$n \neq 2d+1$ $\Rightarrow$ G no es autocomplementario
(8 y 9 salen del ítem a).

---

b - Si G tiene n vértices $\Rightarrow$ para que sea
d-reg., en principio $d \leq n-1$.

Sea $n \in \mathbb{Z}_{\geq 0}$ tq $n \not\equiv 1 \pmod 4$, $n \neq 4k+1$
qvq hay una cantidad par de grafos regulares de
n vértices.

d = 0 1 2 3 4 5 . . . n-2 n-1.
\# 1 1

Idea:
- Si n es par: G_n es d regular $\Leftrightarrow$ G_n^c es
$n-1-d$ reg.
$\Rightarrow$ \# Grafos Reg de n-vert $= \sum_{d=0}^{n-1}$ \#Grafos d-reg.

Incompleto